III. CLAIM AMENDMENTS

1. (Original) A method of managing a plurality of sessions (66) the sessions being between a plurality of terminals (2) and a server (20) having a plurality of threads (74), the method comprising:

grouping the sessions into a plurality of groups (72); and

assigning a thread (74) to each group (72) of sessions so that the assigned thread (74) only handles the events of that group of sessions.

2. (Original) A method according to claim 1 in which grouping occurs when a session is created (70).

3. (Original) A method according to claim 1 in which grouping occurs when a session becomes active.

4. (Previously Presented) A method according to claim 1 in which one group (72) is provided for each thread (74) so that there are equal numbers of groups (72) and threads (74).

5. (Previously Presented) A method according to claim 1 in which sessions are assigned statically to particular threads (74).

6. (Previously Presented) A method according to claim 1 in which a session is put into a first group in a first time period before suspension and put into a second group in a second time period following resumption.

7. (Original) A method according to claim 6 in which the second group is chosen on the basis of the relative levels of activity of the groups.

8. (Original) A method according to claim 6 in which the second group is chosen randomly.

9. (Previously Presented) A method according to claim 1 in which each group (72) has a queue (80) and each session puts its events into that queue (80).

10. (Previously Presented) A method according to claim 1 in which the sessions are grouped by a thread referred to as an acceptor thread (76).

11. (Original) A method according to claim 10 in which the acceptor thread (76) calls a function which is answered by notification that a new session has been created and then assigns the new session to a particular session group (72).

12. (Currently Amended) A method according to claim 1 in which the sessions remain open for an undetermined period of time until closed~~are long-lived~~.

13. (Previously Presented) A method according to claim 1 in which the terminals (2) comprise mobile terminals.

14. (Original) A method according to claim 13 in which the terminals (2) comprise cellular telephones.

15. (Previously Presented) A method according to claim 1 in which load balancing means is included in the assignment mechanism of the session.

16. (Previously Presented) A method according to claim 1 in which the sessions (66) involve obtaining information or conducting transactions through the Internet.

17. (Previously Presented) A method according to claim 1 in which the sessions are part of the Wireless Session Protocol (WSP).

18. (Original) A server (20) for managing a plurality of sessions with a plurality of terminal (2) the server (20) comprising a plurality of threads (74), grouping means to group the sessions into a plurality of groups and assigning means to assign a thread

to each group of sessions so that the assigned thread (74) only handles the events of that group (72) of sessions.

19. (Original) A server (20) according to claim 18 comprising a gateway server serving a plurality of mobile terminals (2).

20. (Original) A server (20) according to claim 19 comprising a WAP-HTTP gateway.

21. (Original) A communications system comprising a server (20) and a plurality of terminals (2) the server (20) and the terminals (2) conducting a plurality of sessions (66) the server comprising a plurality of threads (74), grouping means to group the sessions into a plurality of groups and assigning means to assign at least one thread to each group of sessions so that the assigned thread (74) only handles the events of that group (72) of sessions.

22. (Original) A computer program product for managing a plurality of sessions (66) the sessions being between a plurality of terminals (2) and a server (20) having a plurality of threads (74), comprising:

computer readable program means for grouping the sessions (66) into a plurality of groups (72); and

computer readable program means for assigning a thread to each group (72) of sessions so that the assigned thread (74) only handles the events of that group (72) of sessions.